IN THE CLAIMS

Please amend claims as follows.

1. (Currently Amended) A ~~state machine~~ hardware architecture for the implementation of computational and control flow graphs, the architecture comprising:

a plurality of node elements, wherein each of the plurality of node elements represents a node of a control flow graph;

a plurality of interconnections to connect node elements~~;~~, wherein each interconnection in the plurality of interconnections represents a distinct transition in the control flow graph;

a plurality of ~~state transition~~ connectivity control logic to independently enable and disable each of the connections within the plurality of interconnections to form the control flow graph with the plurality of node elements, the control flow graph being one of multiple possible control flow graphs implementable by enabling and disabling an arbitrary set of the connections within the plurality of interconnections; and

a plurality of ~~state transition~~ evaluation logic elements coupled to the interconnections and operable to evaluate input data against criteria, the plurality of ~~state transition~~ evaluation logic elements to control one or more ~~state~~ transitions between node elements in the control flow graph.

2. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein the criteria comprises evaluation symbols containing specifications for one or more of the group consisting of operations and data.

3. (Currently Amended) The ~~state machine~~ architecture defined in Claim 2 wherein the evaluation symbols are programmable.

4. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 further comprising a data transfer unit through which dynamically computed data is sent to the ~~state transition~~ connectivity control logic.

5. (Cancelled)

6. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein the ~~state transition~~ connectivity control logic comprises a plurality of storage elements, where each storage element in the plurality of storage elements stores a value used to enable or disable an interconnection between node elements in the plurality of interconnections.

7. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein the ~~state transition~~ connectivity control logic is programmable.

8. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein the ~~state transition~~ evaluation criteria are programmable.

9. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein the ~~state transition~~ connectivity control logic and the ~~state transition~~ evaluation symbols are programmable.

10. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 further comprising at least one trigger signal coupled to the plurality of interconnections to gate a ~~state~~ transition across an interconnection in the plurality of interconnections.

11. (Currently Amended) The ~~state machine~~ architecture defined in Claim 10 wherein the at least one trigger signal is generated in response to the set of ~~state transition~~ evaluation logic.

12. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein each of the plurality of node elements comprises a storage element to store a value representing ~~the~~ a state of the control flow graph.

13. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein the set of ~~state transition~~ evaluation logic comprises a plurality of comparators to compare the input data to the criteria.

14. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein the ~~state machine~~ architecture is configured to begin evaluation with a set of programmable start states initialized to be active

15. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein the ~~state machine~~ architecture is configured to perform recognition with an accept state defined to terminate evaluation when reached.

16. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein the plurality of interconnections comprises a plurality of wires.

17. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein the ~~state transition~~ connectivity control logic comprises a plurality of switches.

18. (Currently Amended) The ~~state machine~~ architecture defined in Claim ~~1~~17 wherein switches in the plurality of switches are implemented with logic gates.

19. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein the control flow graph comprises M nodes, M^2 arc transitions and M^2 symbols for evaluation, where M is an integer.

20. (Currently Amended) The ~~state machine~~ architecture defined in Claim 19 wherein M is one of a group consisting of 16 and 32.

21. (Currently Amended) The ~~state machine~~ architecture defined in Claim 1 wherein the control flow graph comprises M nodes, M^2 are transitions and M symbols for evaluation, where M is an integer.

22. (Currently Amended) The ~~state machine~~ architecture defined in Claim ~~20~~ 21 wherein M is one of a group consisting of 16 and 32.

23. (Cancelled)

24. (Canceled)

25. (Currently Amended) An ~~state machine~~ architecture for the implementation of computational and control flow graphs, the architecture comprising:

a plurality of supernodes,

a plurality of interconnections to connect supernode elements;

a plurality of ~~state transition~~ connectivity control logic to enable and disable connections within the plurality of interconnections to form ~~the~~ a first control flow graph with the plurality of node elements; and

a plurality of ~~state transition~~ evaluation logic coupled to the interconnections and operable to evaluate input data against criteria, the plurality of ~~state transition~~ evaluation logic to control one or more state transitions between supernode elements in the first control flow graph,

wherein one of the plurality of supernodes comprises a ~~state machine~~ control flow graph having

a plurality of node elements, wherein each of the plurality of node elements represents a node of a second control flow graph;

a plurality of interconnections to connect node elements;

a plurality of ~~state transition~~ connectivity control logic to enable and disable connections within the plurality of interconnections to form ~~the~~ a second control flow graph with the plurality of node elements; and

a plurality of ~~state transition~~ evaluation logic coupled to the interconnections and operable to evaluate input data against criteria, the plurality of ~~state transition~~ evaluation logic to control one or more ~~state~~ transitions between node elements in the second control flow graph.

26. (New) A hardware architecture for implementing computational and control flow graphs for realization of programmable non-deterministic Finite State Automata to evaluate regular expressions, wherein the control flow graph comprises of M nodes, M^2 arc transitions, M^2 symbols for evaluation and M^2 storage elements, where M is an integer, and wherein the M nodes are potentially fully connectable to each other using the M^2 arc transitions, wherein each of the M^2 storage elements stores a value used to independently enable or disable an associated one of the M^2 arc transitions between the M nodes, and wherein the input data is compared against M^2 symbols to determine whether control transitions across the M^2 arcs should exist.

27. (New) The hardware architecture defined in Claim 26 wherein the input data is compared against M symbols organized as one symbol per node to control interconnections to or from that node, wherein comparisons against the M symbols are used to determine whether control transitions across the M^2 arcs should exist.